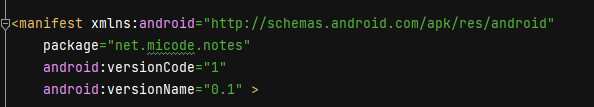
**AndroidManifest.xml**

每一个android应用程序项目都必须包含有AndroidManifest.xml文件（后称清单文件）并且放在项目资源集的根目录中（即main文件夹下）。清单文件描述了应用必要的信息提供给Android编译工具、Android操作系统以及应用商店（如Google Play）。

### **包名和应用ID（application ID）**

清单文件的根元素要求声明应用的包名称，一般情况下与项目中的Java目录结构一致。



把应用打包成最终的APK，Android编译工具将 package属性应用于以下2处：

* 将该名称作为应用的命名空间以生成.java类文件
* 用以解决清单文件中的相似类名，以缩短命名。

### **设备兼容性(Device compatibility)**

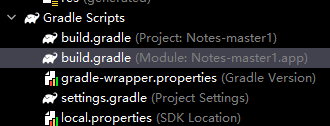
清单文件中可以声明应用要求软硬件的特性以及应用兼容哪一种类型的设备。Google 应用商店不允许应用安装在与系统版本、特性不匹配的设备上。

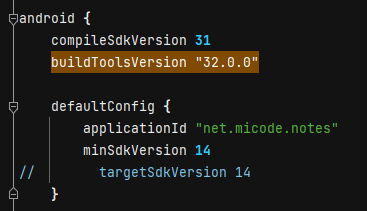
在这次的小米便签源码中，关于设备兼容性部分只是声明了sdk版本。



每次发布的平台系统版本，通常会添加一些之前版本不兼容的新的API。为了表明应用程序所兼容的最低版本，必须在<uses-sdk>标签中声明minSdkVersion属性值。

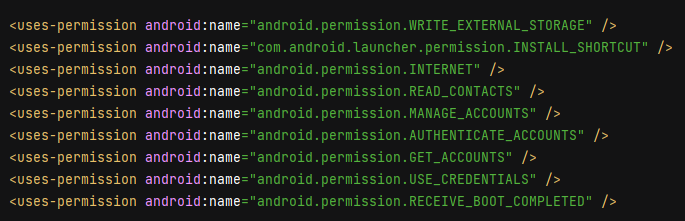
需要注意的是，<uses-sdk>由build.gradle文件中的相对应的属性重写。因此如果使用Android Studio，就必须在build.gradle文件中定义minSdkVersion和targetSdkVersion值。





### **权限(Permissions)**

Android应用如果要访问用户的敏感数据（例如联系人和短信）或者系统功能（如摄像头和网络）就必须申请相应的权限并用每个权限有独特的标签进行标识。



可以看到小米便签申请的权限有：

* 写入外部存储
* 添加快捷方式
* 访问网络
* 读取联系人
* 管理账户
* 验证账户
* 访问账户Gmail列表
* 使用证书
* 开机自启动

### **Icons和labels**

大多数的manifest节点都有icon和label属性来显示一个小图标和文本标签，以引导用户找到对应的应用组件。

但凡在父节点中声明了icon和label值，该值将会变成子节点的默认值。比如在<application>节点声明该值作为Activity的默认值。

line36：

android:icon="@drawable/icon\_app"

指向res/drawable/icon\_app 的app图标

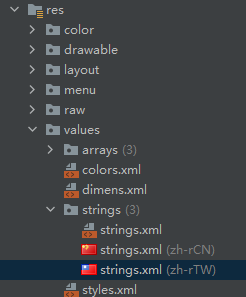
如下



line37：

android:label="@string/app\_name" >

指向res/values/strings/strings.xml



定义如下：

<string name="app\_name">Notes</string>

<string name="app\_name">便签</string>

<string name="app\_name">便簽</string>

之后我们介绍一下在这个文件中注册的第一个activity组件，这是应用的启动入口页面，当用户点击应用图标，该Activity会被启动。

line39-line51：

***<activity***

***android:name=".ui.NotesListActivity"***

***android:configChanges="keyboardHidden|orientation|screenSize"***

***android:label="@string/app\_name"***

***android:launchMode="singleTop"***

***android:theme="@style/NoteTheme"***

***android:uiOptions="splitActionBarWhenNarrow"***

***android:windowSoftInputMode="adjustPan" >***

***<intent-filter>***

***<action android:name="android.intent.action.MAIN" />***

***<category android:name="android.intent.category.LAUNCHER" />***

***</intent-filter>***

***</activity>***

其中

***android:name***

指向对应路径。此处指向的***NotesListActivity***类正是软件启动后的第一个界面。

***android:configChanges***

所列出的状态发生变更时候，Activity会调用onConfigurationChanged()方法

三个参数分别为：

***keyboardHidden***

键盘无障碍功能发生了变化 — 例如，用户显示了硬件键盘。

***orientation***

屏幕方向发生了变化 — 用户旋转了设备。

***screenSize***

当前可用屏幕尺寸发生了变化。它表示当前可用尺寸相对于当前纵横比的变化，因此会在用户在横向与纵向之间切换时发生变化。

此处代码的作用为，在使用软键盘，改变屏幕方向和改变屏幕尺寸时，不再去执行onCreate()方法，而是直接执行onConfigurationChanged()。如果不申明此段代码，按照Activity的生命周期，都会去执行一次onCreate()方法，而onCreate（）方法通常会在显示之前做一些初始化工作。所以如果改变屏幕方向这样的操作都去执行onCreate()方法，就有可能造成重复的初始化，降低程序效率,而且更有可能因为重复的初始化而导致数据的丢失。

***android:label***

会在屏幕上显示的Activity标签，通常和上面的icon一起出现

***android:launchMode***

用来设置android的启动方式，此时参数为

***singleTop***

如果当前Activity处于栈顶，那么直接复用，否则重新启动Activity

***android:theme***

设置Activity的主题。指向res/values/styles.xml，定义如下

**<style name="NoteTheme" parent="@android:style/Theme.Holo.Light">**

**<item name="android:actionBarStyle">@style/NoteActionBarStyle</item>**

**</style>**



***android:uiOptions***

针对ActivityUI的附加选项。参数为：

***splitActionBarWhenNarrow***

当水平空间受限制，在屏幕底部添加一栏，以显示应用栏中的操作项。

***android:windowSoftInputMode***

Activity 的主窗口与包含屏幕软键盘的窗口的交互方式。 该属性的设置影响两个方面：

* 当 Activity 成为用户注意的焦点时软键盘的状态 — 隐藏还是可见。
* 对 Activity 主窗口所做的调整 — 是否将其尺寸调小以为软键盘腾出空间，或者当窗口部分被软键盘遮挡时是否平移其内容以使当前焦点可见。

参数为

***adjustPan***

不调整 Activity 主窗口的尺寸来为软键盘腾出空间， 而是自动平移窗口的内容，使当前焦点永远不被键盘遮盖，让用户始终都能看到其输入的内容。 这通常不如尺寸调正可取，因为用户可能需要关闭软键盘以到达被遮盖的窗口部分或与这些部分进行交互。

### **意图过滤器（Intent filters）**

应用的Activity、Service、Broadcast Receiver、由intent激活。intent是由Intent对象定义的一个用以描述动作与行为的消息。  
包括可用的数据、可表示行为的组件类别以及其他指令。

当应用分发一个intent给操作系统，操作系统可以通过应用清单文件的*intent filter*定位到可接收该intent应用组件。如果超过了一个应用可以处理该intent，系统会弹出选择对话框让用户选择使用哪个应用进行处理。

应用组件拥有无数个*intent filter*(使用进行定义)，每个都可以声明组件的不同功能。

line47-line50：

***<intent-filter>***

***<action android:name="android.intent.action.MAIN" />***

***<category android:name="android.intent.category.LAUNCHER" />***

***</intent-filter>***

* android.intent.action.MAIN，表示该Activity是应用的启动入口，且不需要任何Intent携带的数据。
* android.intent.category.LAUNCHER，表示将该Activity的图标设为手机主屏幕上的应用图标，如果它没有图标，就用Application的图标。

参考：

1. [Activity详解——Activity的xml配置](https://blog.csdn.net/h55l55/article/details/80350846)
2. [AndroidManifest.xml 文件详解](https://blog.csdn.net/u013560890/article/details/81240356)
3. [Intent以及IntentFilter详解](https://blog.csdn.net/xiaohanluo/article/details/52637520)
4. [Android各权限说明汇总](https://www.jianshu.com/p/665e67c1cdbe)